Remarks

Reconsideration of this application in light of the remarks below and allowance of all pending claims are respectfully requested. Claims 45-64 remain pending.

In the Office Action, dated January 13, 2006, claims 45-64 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,263,489 to Olsen et al. in view of U.S. Patent No. 6,282,698 to Baker et al. Applicant respectfully, but most strenuously, traverses this rejection for the reasons herein.

In one aspect, applicant's invention is directed to automatically restoring debugging breakpoints subsequent to modifying the source code of a program. The automatically restoring restores the breakpoint to the same step the breakpoint was set prior to modification of the source code, although that step is now at a different location within the modified program.

As an example, to automatically restore the debugging breakpoint to the same step of the program that had the breakpoint prior to modification, the location (e.g., line of code) that includes the selected step is determined. This determination is made by creating an instruction profile that includes attributes of a number of generated instructions that are associated with the selected step. These attributes include attributes of the instruction (such as, an operation code, operand and operation type obtained from reading the instructions) of the selected step, and possibly attributes of other instructions in proximity to the selected step.

When the program is modified, the attributes of the instruction profile are compared to the attributes of the newly generated instructions to determine where the particular line of code of interest has moved in the program. The various comparisons made to different instructions within the modified program yield difference counters. In this example, the difference counter having the smallest value indicates the location of the selected step. Thus, the debugger automatically restores the breakpoint to that step.

In one particular example, applicant claims a method of restoring debugging breakpoints (e.g., independent claim 1). The method includes, for instance, having a breakpoint that is set to a selected step of a first version of source code of a program, the

program being absent embedded debug commands; creating an instruction profile for the selected step, the instruction profile including one or more attributes of one or more machine instructions generated for the selected step and one or more attributes of zero or more other machine instructions generated for the first version of source code; and automatically restoring the breakpoint to the selected step of a modified program, in response to modification of the first version of source code to provide the modified program having a second version of source code, wherein the selected step is at a different location within the modified program, and wherein the automatically restoring includes comparing one or more attributes of the one or more machine instructions generated for the second version of source code with one or more attributes of the instruction profile created based on the first version of source code to determine the different location.

Thus, in this aspect of applicant's claimed invention, a breakpoint is automatically restored to the selected step of a modified program, in response to modification of the source code of the program. Further, the restoring compares attributes of one or more machine instructions generated for a second version of source code with one or more attributes of an instruction profile created based on a first version of source code to determine the new location for the breakpoint. These features are very different from the teachings of Olsen and Baker, either alone or in combination.

In Olsen, while a technique is described for debugging optimized code, Olsen fails to describe, teach or suggest one or more aspects of applicant's claimed invention. For instance, there is no description, teaching or suggestion in Olsen of automatically restoring a breakpoint, in response to modification of a first version of the source code to provide a modified program having a second version of the source code. There is no description in Olsen of modifying the source code. Instead, Olsen is directed to problems associated with optimized code and does not address handling different versions of source code. This failure of Olsen is explicitly admitted in the Office Action (see, e.g., page 4 of the Office Action).

Further, Olsen fails to describe, teach or suggest applicant's claimed element of comparing one or more attributes of one or more machine instructions generated for the second version of the source code with one or more attributes of the instruction profile created based on a first version of the source code to determine the location in which to

restore the breakpoint after modification of the source code. Again, the failure of Olsen to teach or suggest this aspect of applicant's claimed invention is explicitly admitted in the Office Action. Thus, Baker is relied upon. However, Baker fails to overcome the deficiencies of Olsen.

Baker is directed to determining whether binary programs are similar to one another. Baker is not at all concerned with debugging programs or the restoring of debugging breakpoints, as claimed by applicant. Thus, Baker fails to teach or suggest one or more of applicant's claimed elements. For instance, Baker fails to describe, teach or suggest applicant's claimed element of automatically restoring the breakpoint to the selected step of a modified program, in response to modification of the first version of source code to provide the modified program having the second version of source code. Baker fails to describe the restoring of a breakpoint at all. Baker simply determines whether two programs are similar. There is no discussion in Baker of debugging or of breakpoints, and there is certainly no discussion in Baker of automatically restoring the breakpoint, as claimed by applicant. Since both Olsen and Baker fail to describe, teach or suggest this same aspect of applicant's claimed invention, applicant respectfully submits that the combination of Baker and Olsen also fails to teach or suggest this aspect of applicant's claimed invention.

Further, applicant respectfully submits that Baker fails to describe, teach or suggest applicant's claimed feature of "wherein the automatically restoring comprises comparing one or more attributes of one or more machine instructions generated for the second version of source code with one or more attributes of the instruction profile created based on the first version of source code to determine the different location of the selected step." Baker is not concerned with where a selected step is in the second version of the program, just that the two versions are similar. Since both Baker and Olsen fail to teach or suggest this claimed feature, applicant respectfully submits that the combination fails to teach or suggest this claimed feature.

The combination of the references (*assuming arguendo* the combination is proper) merely teaches that a determination may be made as to whether two binaries are similar and that a breakpoint can be restored in optimized code. There is no teaching or suggestion in the combination of automatically restoring a breakpoint in a modified program having a second

version of source code, as claimed by applicants. In particular, there is no teaching or suggestion in the combination of automatically restoring the breakpoint to a selected step of a modified program, in response to modification of the first version of source code to provide the modified program having a second version of source code. The determination that two binaries are similar is not a teaching of what is claimed. It is not a teaching at all of restoring a breakpoint. Baker does not help applicant address the problem applicant was trying to solve. Applicant already knew that the programs were similar – one was a modified version of the other. What applicant was trying to solve was automatically restoring a breakpoint to a selected step that was in a different location in a second version of source code. This is not taught by Baker, which simply provides a way to determine whether two binaries are similar, something applicant already knew. Further, it is not taught by Olsen, which describes restoring optimized code. There is no teaching in the combined references of how one would automatically restore a breakpoint to a selected step of a modified program having a second version of source code. Thus, applicant respectfully submits that the combination fails to describe, teach or suggest applicant's claimed invention.

In addition to the above, applicant respectfully submits that both Olsen and Baker fail to describe, teach or suggest applicant's claimed element of having a breakpoint that is set to a selected step of a first version of source code of a program, the program being absent embedded debug commands. While Olsen may set a breakpoint, Olsen's program is not absent embedded debug commands. In Olsen, debug information is generated by the compiler and included within the program. This is described in col. 8, lines 40-50 of Olsen. In the Office Action, it is indicated that col. 4, lines 20-30 of Olsen show a portion of source code with no debug commands in that portion. Applicant respectfully submits that even though this portion is shown without debug commands, it is clearly stated in Olsen that the program will have debug commands therein. Olsen expressly teaches adding debugging information to the program during the compilation stage. This debugging information is generated by the compiler to provide information vital to providing the highwater (HW) and lowwater (LW) points. The mapping of machine code instructions to source constructions is necessary to generate the HW and LW points for a given source line (col. 9, lines 23-23). To identify an instruction that corresponds to the source construct requires the mapping table generated by the compiler. Therefore, applicant respectfully submits that Olsen does not

teach or suggest applicant's claimed invention of having a breakpoint that is set to a selected step of a program, that program being absent embedded debug commands. Further, Baker does not overcome the deficiencies of Olsen, since Baker does not set a breakpoint at all. Since both references fail to teach or suggest this aspect of applicant's claimed invention, the combination also fails to teach or suggest this aspect of applicant's claimed invention.

For at least the above reasons, applicant respectfully submits that the combination of Olsen and Baker fails to describe, teach or suggest one or more aspects of applicant's claimed invention. Additionally, applicant respectfully submits that the combination of Olsen and Baker is improper.

Applicant respectfully submits that the combination of Olsen and Baker is improper, since at the very least, there is no teaching or suggestion in the references themselves to make the combination or modification suggested in the Office Action. It is well known that:

> It is insufficient to establish obviousness that the separate elements of the invention existed in the prior art; absent some teaching or suggestion, in the prior art to combine the elements. Arkie Loures Inc. v. Gene Larew Tackle Inc., 43 U.S.P.Q. 2d 1294, 1297 (Fed. Cir. 1997).

Applicant respectfully submits that there is no such teaching or suggestion in the references. Any justification for the combination provided in the Office Action does not indicate where the references expressly teach the combination. Instead, the combination or suggested modification appears to be a hindsight reconstruction of applicant's invention. That is, the justification is simply a selection of various elements of the combination in an attempt to create applicant's invention, rather than a reason for the combination drawn from the references or from the knowledge available to one of ordinary skill in the art.

Further, Olsen and Baker are each trying to solve different problems, and are not reasonably pertinent to the particular problem with which the inventor is involved. Baker has no relation to Olsen. Olsen is concerned with setting breakpoints in an optimized version of source code. In Olsen, there is only one version of source code. There is no discussion at all in Olsen of wanting or needing multiple versions of the source code. Olsen is merely concerned with debugging optimized code of one version of source code. In contrast, Baker is concerned with multiple versions of code and not at all with debugging the code or setting

breakpoints. Baker is directed to determining whether two binaries are similar. Thus, Olsen and Baker are directed at different problems and would not be combined. Further, they are not directed to applicant's problem, which includes determining how to automatically restore a breakpoint of a modified program having a second version of source code. Baker is not even concerned with debugging at all.

Based on the foregoing, applicant respectfully submits that the combination of Olsen and Baker is improper, and therefore independent claim 1 is patentable over the combination. Further, applicant respectfully submits that even if the combination is proper, that the combination fails to teach or suggest one or more of applicant's claimed elements. Therefore, independent claim 1 and the other independent claims are patentable over the combination.

Additionally, applicant respectfully submits that the dependent claims are patentable for the same reasons as the independent claims, as well as for their own additional features. For instance, applicant explicitly recites in dependent claims 47, 55 and 61 that the instruction profile includes a source line number for the selected step and the length of the first version of source code and that the automatically restoring uses these values to determine a starting point within the modified program to select the one or more instructions generated for the second version to be used in the comparing. This is not described, taught or suggested in Olsen or Baker, either alone or in combination.

In Olsen, high watermarks and low watermarks are used to determine which lines of code are to be emulated. However, there is no description, teaching or suggestion of using the source line number of a first version of the source code and a length of the first version of source code to determine a starting point from which one or more instructions generated for the second version of source code are to be used in the comparing in order to automatically restore the breakpoint to the selected step within the modified program.

Further, Baker does not overcome the deficiencies of Olsen. There is no teaching or suggestion in Baker of the above claimed elements. Applicant respectfully submits that this is even implicitly admitted in the Office Action, since the Office Action does not give any indication of how Baker overcomes the deficiencies of Olsen. Thus, applicant respectfully

submits that dependent claims 47, 55 and 61 are patentable over the combination of Olsen and Baker.

In yet a further example, applicant recites in dependent claims 51, 58 and 64 a stepwise procedure for choosing a number of instructions to be included in the instruction profile used in the comparing step. Applicant respectfully submits that this stepwise procedure, which includes, for instance, selecting a number of instruction to be included in the calibration profile; generating the calibration profile for a chosen line of the program, the calibration profile having the selected number of instructions for the chosen line; comparing one or more attributes of the calibration profile to one or more attributes of at least one line of code of the program to obtain end results; determining whether the result is an unambiguous result; and repeating, zero or more times, the selecting, the generating, the comparing and the determining until the determining indicates an unambiguous result, wherein the selected number of instructions increases at each iteration, and wherein the selected number of instructions indicates when there is an indication of an unambiguous result, the number of machine instructions to be included in the instruction profile, is not described, taught or suggest in Olsen or Baker, either alone or in combination.

For all of the above reasons, applicant respectfully submits that all claims pending herein are patentable over the combination of Olsen and Baker.

Should the Examiner wish to discuss this case with applicant's attorney, please contact applicant's attorney at the below listed number.

Respectfully submitted,

Blanche E. Schiller

Blanche E. Schiller
Attorney for Applicant
Registration No.: 35,670

Dated: April 7, 2006.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579